Machine Learning Final Project:

Imbalance Classification

Course: Machine Learning I

Instructor: Dr. Tariq Mahmood

Submitted by: Saad Ullah Bilal (29416) and Muhammad Zain (29403)

The target of this project is to discover the impact of using different techniques to address class imbalance (CI) on ML performance

Brief overview of what each part of my execution pipeline is doing

Overview:

The Master() function acts as the central command center of the pipeline, managing various tasks such as data loading, imbalance checking, data cleaning, transformation, exploratory data analysis (EDA), application of balancing techniques, PCA, feature selection, and model training and evaluation.

Here is a Breakdown:

1. **Loading Data:**
   a) **load_dataset(file_path, id):**
      - **Inputs:** Can be a path to a CSV file (file_path) or a dataset ID to fetch from the UCI repository.
      - **Outputs:** Prints initial data insights such as column names and the count of columns loaded.
2. **Checking for Imbalance:**
   a) check_imbalance(data, 'target'):
      - **Functionality:** Examines the class distribution of the target column.
      - **Outputs:** If significant imbalance is detected, it displays a bar plot of class distribution and the count of instances per class, aiding in the decision of whether balancing techniques are required.
3. **Data Cleaning:**
   a) **clean_data(data, encode_target=False):**
      - **Functionality:** Standardizes numerical features and imputes missing values using the mean for numerical columns. Optionally encodes the

target column if encode_target is set to True, using one-hot encoding for categorical features.

- **Outputs:** Returns the cleaned and optionally encoded dataset.

4. **Converting Target Variable:**
   a) **convert_target_to_categorical(data, 'target', bins=3):**
      - **Functionality:** If the target is numerical, discretizes the variable into specified bins (default is 3), preparing it for classification.
5. **Exploratory Data Analysis (EDA):**
   a) **perform_eda(data):**
      - **Functionality:** Provides comprehensive insights into the data, including structure, summary statistics, distribution plots, box plots, and correlation matrices of features most correlated with the target.
      - **Outputs:** Helps understand the dataset's nature and potentially predictive features.
6. **Applying Balancing Techniques:**
   a) **Balancing Techniques:** Utilized to address class imbalance.
      - **model_rf(data. copy(), 'target'):** Uses a Random Forest model with adjusted class weights to balance the data.
      - **smote(data. copy(), 'target'):** Applies SMOTE technique to oversample the minority class**.**
      - **cc(data. copy(), 'target'):** Uses Cluster Centroids for under sampling the majority class.
      - **svm(data. copy(), 'target'):** Demonstrates the application of SVM for anomaly detection.
      - **Effects:** Each technique modifies the dataset, potentially affecting subsequent model training and evaluation.
7. **Feature Selection and Reduction:**
   a) **Post-Balancing Operations:**
      - **apply_pca(X):** Reduces dimensionality while retaining maximum variance.
      - **feature_selection(X_pca, y):** Uses a RandomForest model to perform feature selection and isolate the most important features.
8. **Model Training and Predictive Analysis:**
   a) **train_and_predict(X_selected, y, classifiers):**
      - **Functionality:** Fits several classifiers on the selected features of balanced datasets.

- **Evaluations:** Each classifier model is evaluated through metrics such as ROC AUC, F1 Score, Precision, Recall, and Accuracy. Detailed classification reports provide deeper insights into each model.

9. **Results Compilation:**
   a) **Functionality:** Compiles results from multiple models and balancing techniques into a tabulated format for easy comparison.
   b) **Objective:** To identify which techniques or combinations yield the best results.

Execution Flow: The data is processed sequentially through each function stage, culminating in a comprehensive report featuring performance metrics across different models and data balancing strategies. This modular approach allows for the isolation of each technique's impact, optimizing the machine-learning pipeline for optimal performance on potentially imbalanced datasets.

Now, I'll discuss the performance of my model with different balancing techniques with different classification models

Source for first Dataset: https://archive.ics.uci.edu/dataset/1/abalone

This dataset is about the **Prediction of the age** of **abalone** from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

Machine Learning Final Project:

Imbalance Classification

Top model

Random Forest (SMOTE balanced):

- **Accuracy** = 0.799756,
- **ROC AUC** = 0.942438,
- **F1 Score** = 0.

This model performs well, as we can observe that in every metric, the ROC AUC approaches 0.94, which is able to make a clear discrimination between classes.

Why Random Forest with SMOTE is the Best Balanced Performance:

- **Balanced Performance**
  The high ROC AUC, the F1 score, precision, and recall are balanced, indicating that the model gives good results in giving out the right cases of positive classes and, at the same time, minimizing false positives.
- **Appropriate Complexity:**
  This model does not reach the high level of accuracy or the highest ROC AUC that the more complex models do. The best model in terms of ROC AUC, Cluster Centroids with Gradient Boosting, only achieved suspiciously high metrics that most probably were indicative of overfitting.
- **Imbalanced-data-friendly fit:**
  SMOTE (Synthetic Minority Over-sampling Technique) is applied as a means to treat class imbalance, which is rather common in practicality, thus making the model really generally applicable.

Machine Learning Final Project:

Imbalance Classification

Source for Second Dataset: https://www.kaggle.com/datasets/mexwell/heart-disease-dataset

This dataset is about the **prediction of heart disease** consists of 1190 instances with 11 features. These datasets were collected and combined at one place to help advance research on CAD-related machine learning and data mining algorithms, and hopefully to ultimately advance clinical diagnosis and early treatment.

Top Model Selections:

1) **Gradient Boosting (Cluster Centroids)**
    - **Accuracy:** 88.324%
    - **ROC AUC:** 95.969%
    - **F1 Score:** 88.321%
    - **Precision:** 88.368%
    - **Recall:** 88.324%

    Justification: This model offers high ROC AUC, indicating strong discriminative ability, along with balanced precision and recall. The F1 Score is close to accuracy, suggesting a balanced performance on both positive and negative classes.

2) **K-Nearest Neighbors (SMOTE)**
    - **Accuracy:** 87.202%
    - **ROC AUC:** 95.607%
    - **F1 Score:** 87.198%
    - **Precision:** 87.252%
    - **Recall:** 87.202%

    Justification: Offers nearly balanced metrics across the board with a high ROC AUC, which means good classification effectiveness. This model shows robustness against class imbalance due to the effective use of SMOTE.

3) **Gradient Boosting (SMOTE)**
    - **Accuracy:** 87.917%
    - **ROC AUC:** 95.633%
    - **F1 Score:** 87.909%
    - **Precision:** 88.028%
    - **Recall:** 87.917%

    Justification: Similar to the K-Nearest Neighbors with SMOTE, this model shows good generalization capability with high ROC AUC and balanced accuracy, F1, precision, and recall. Gradient Boosting with SMOTE also handles class imbalance effectively.

Recommendation

The Gradient Boosting model balanced with Cluster Centroids is the recommended best choice for better performance. Having the highest ROC AUC among models indicating less than 90% accuracy, one is said to represent an excellent ability to differentiate between the class outcomes. The metrics are pretty balanced and show quite stable performance at different dimensions of the model's predictive power.

This model has obtained a nice balance in terms of not overfitting, while at the same time supporting a high level of predictive power—thus, adequate for a solid generalization over similar datasets.

Source for third Dataset: https://www.kaggle.com/datasets/aadarshvelu/aids-virus-infection-prediction

This Dataset contains healthcare statistics and categorical information about patients who have been diagnosed with AIDS. This dataset was initially published in 1996. The learning will show if the **patients will have AIDS or not** based on the several parameters

Machine Learning Final Project:

Imbalance Classification

Top Model Selections:

1) **K-Nearest Neighbors (One-Class SVM)**
   - **Accuracy:** 76.74
   - **ROC AUC:** 81.659%
   - **F1 Score:** 75.434%
   - **Precision:** 75.684%
   - **Recall:** 76.74%
   - **Justification:** This model offers the highest accuracy among those below 90%, with a reasonably high ROC AUC, suggesting good discriminative capability. The balance between precision and recall indicates effective classification without bias toward either class.
2) **K-Nearest Neighbors (Cluster Centroids)**
   - **Accuracy:** 73.652%
   - **ROC AUC:** 81.171%
   - **F1 Score:** 73.652%
   - **Precision:** 73.652%
   - **Recall:** 73.652%
   - **Justification:** Consistent performance across all metrics with a good ROC AUC. This model demonstrates an ability to manage class imbalance effectively, using the Cluster Centroids method.
3) **K-Nearest Neighbors (SMOTE)**
   - **Accuracy:** 73.559%
   - **ROC AUC:** 81.221%
   - **F1 Score:** 73.555%
   - **Precision:** 73.573%
   - **Recall:** 73.559%
   - **Justification:** Similar to the K-Nearest Neighbors with Cluster Centroids, this model has balanced metrics and is effective in handling imbalanced data through the use of SMOTE.

Final Recommendation:

The **K-Nearest Neighbors (One-Class SVM)** is recommended as the best choice. This model achieves the highest accuracy among those considered while maintaining a solid

ROC AUC score, which signifies strong capability in differentiating between classes. Additionally, the balance in precision and recall shows that the model is neither too conservative nor too aggressive in predicting the positive class, making it suitable for scenarios where both false positives and false negatives carry significant costs.

Source for fourth Dataset: https://www.kaggle.com/datasets/ahsan81/superstore-marketing-campaign-dataset

The **superstore** wants to **predict the likelihood of the customer** giving a **positive response** and wants to identify the different factors which affect the customer's response. You need to analyze the data provided to identify these factors and then build a prediction model to predict the probability of a customer will give a **positive response.**

Top Model Selections:

1) **K-Nearest Neighbors (SMOTE)**
   - **Accuracy:** 87.539%
   - **ROC AUC:** 96.222%
   - **F1 Score:** 87.482%
   - **Precision:** 88.246%
   - **Recall**: 87.539%
   - **Justification:** This model has the highest ROC AUC among those with accuracy below 90%. It indicates a strong ability to distinguish between classes and maintains good precision and recall balance, making it robust for varied data distributions.
2) **Gradient Boosting (Cluster Centroids)**
   - **Accuracy:** 91.317%
   - **ROC AUC:** 98.105%
   - **F1 Score:** 91.316%
   - **Precision:** 91.341%
   - **Recall:** 91.317%

- **Justification:** While this model slightly exceeds the 90% accuracy threshold, it presents an exceptional balance of metrics across the board. If a slight potential for overfitting is acceptable, this could be considered based on its performance.

3) **Gradient Boosting (SMOTE)**
   - **Accuracy:** 83.054%
   - **ROC AUC:** 90.883%
   - **F1 Score:** 83.049%
   - **Precision:** 83.089%
   - **Recall:** 83.054%
   - **Justification:** Provides robust performance with decent ROC AUC, suggesting good generalization capabilities. The precision and recall are well-balanced, making it a solid choice for handling imbalanced datasets.

Final Recommendation:

The K-Nearest Neighbors (SMOTE) is recommended as the best choice among the models with accuracy below 90%. This model provides a high ROC AUC, which is crucial for ensuring the model's ability to distinguish between classes effectively. Additionally, the precision and recall scores are well-balanced, indicating that the model does not overly favor one class over the other, which is particularly important in practical applications where both false positives and false negatives have significant implications.

Overall Assessment

The performance of classification algorithms combined with different balancing techniques reveals varied effectiveness across scenarios.

Gradient Boosting and K-Nearest Neighbors generally show strong performance, particularly with SMOTE and Cluster Centroids, indicating good capability in managing imbalanced datasets.

Random Forest tends to show signs of overfitting with nearly perfect metrics across most balancing methods.

Logistic Regression and SVM show moderate success, often struggling with lower ROC AUC scores, suggesting limitations in probability estimation and class separation in imbalanced contexts.